Chapter 19
Configure SONET/SDH Interfaces

Synchronous Digital Hierarchy (SDH) is a CCITT standard for a hierarchy of optical transmission rates. Synchronous Optical Network (SONET) is a USA standard that is largely equivalent to SDH. Both are widely used methods for very high speed transmission of voice and data signals across the numerous world-wide fiber-optic networks.

SDH and SONET use light-emitting diodes or lasers to transmit a binary stream of light-on and light-off sequences at a constant rate. At the far end optical sensors convert the pulses of light back to electrical representations of the binary information.

In Wavelength Division Multiplexing (WDM), light at several different wavelengths (colors to a human eye) is transmitted on the same fiber segment, greatly increasing the throughput of each fiber cable.

In Dense Wavelength-Division Multiplexing (DWDM), many optical data streams at different wavelengths are combined into one fiber.

The basic building block of the SONET/SDH hierarchy in the optical domain is an OC-1; in the electrical domain, it is an STS-1. An OC-1 operates at 51.840 Mbps. OC-3 operates at 155.520 Mbps.

A SONET stream can consist of discrete lower-rate traffic flows that have been combined using Time-Division Multiplexing (TDM) techniques. This method is useful, but a portion of the total bandwidth is consumed by the TDM overhead. When a SONET stream consists of only a single, very high speed payload, it is referred to as operating in concatenated mode. A SONET interface operating in this mode has a “c” added to the rate descriptor. For example, a concatenated OC-48 interface is referred to as OC-48c.

SONET and SDH traffic streams exhibit very few differences in behavior that are significant to Juniper Networks SONET/SDH interfaces; in general, this chapter uses *SONET/SDH* to indicate behavior that is identical for the two standards. However, there is one important difference that requires you to configure the interface specifically for SONET or SDH mode. That difference is in the setting of two bits (the ss-bits) in the pointer. SONET equipment ignores these bits, but SDH equipment uses them to distinguish a VC-4 payload from other types. When configured in SDH mode, Juniper Networks SONET/SDH PICs set the ss-bits to `s1s0 2` (binary 10). For more information, see the *JUNOS Internet Softw are Guide: Getting Started*.

This chapter discusses the following SONET/SDH interface properties that you can configure:

Configure SONET/SDH Physical Interface Properties

To configure SONET/SDH physical interface properties, include the sonet-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces so-fpc/pic/port]
sonet-options {
    aggregate asx;
    aps {
        advertise-interval milliseconds;
        authentication-key key;
        force;
        hold-time milliseconds;
        lockout;
        neighbor address;
        paired-group group-name;
        protect-circuit group-name;
        request;
        revert-time seconds;
        working-circuit group-name;
    }
    bytes {
        e1-quiet value;
        f1 value;
        f2 value;
        s1 value;
        z3 value;
        z4 value;
    }
    fcs (32 | 16);
    loopback (local | remote);
    path-trace trace-string;
    (payload-scrambler | no-payload-scrambler);
    rfc-2615;
    (z0-increment | no-z0-increment);
}
```

Note that when you configure SONET/SDH OC-48 interfaces for channelized (multiplexed) mode (by including the no-concatenate statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level), the bytes e1-quiet and bytes f1 options have no effect. The bytes f2, bytes z3, bytes z4, and path-trace options work correctly on channel 0 and work in the transmit direction only on channels 1, 2, and 3. When using no-concatenate, you must specify a channel. For more information, see the *JUNOS Internet Software Guide: Getting Started*.

For DS-3 channels on a Channelized OC-12 interface, the bytes e1-quiet, bytes f1, bytes f2, bytes z3, and bytes z4 options have no effect. The bytes s1 option is supported only for channel 0; it is ignored if configured on channels 1 through 11. The bytes s1 value configured on channel 0 applies to all channels on the interface.

You also can include some of the statements in the sonet-options statement to set SONET parameters on ATM interfaces.

You can configure the following SONET/SDH physical interface properties:

Configure SONET Header Byte Values

To configure values in SONET header bytes, include the bytes statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
user@host# show
bytes {
    e1-quiet value;
    f1 value;
    f2 value;
    s1 value;
    z3 value;
    z4 value;
}
```

You can configure the following SONET header bytes:

e1-quiet—Default idle byte sent on the orderwire SONET overhead bytes. The router does not support the orderwire channel, and hence sends this byte continuously. For the E1-quiet byte, *value* can be in the range 0 through 255. The default value is 0x7F.

f1, f2, z3, z4—SONET overhead bytes. For these bytes, *value* can be in the range 0 through 255. The default value is 0x00.

s1—Synchronization message SONET overhead byte. This byte is normally controlled as a side effect of the system reference clock configuration and the state of the external clock coming from an interface if the system reference clocks have been configured to use an external reference. For the s1 byte, *value* can be in the range 0 through 255.

On SONET OC-48 interfaces that you configure for channelized (multiplexed) mode (by including the no-concatenate statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level), the bytes e1-quiet and bytes f1 options have no effect. The bytes f2, bytes z3, bytes z4, and path-trace options work correctly on channel 0 and work in the transmit direction only on channels 1, 2, and 3. Table 20 displays JUNOS software framing bytes for the stated speeds:

1XSTM-4 is the STM-4/OC-12 PIC

4XSTM-4 is the STM-16/OC-48 PIC set to no-concatenate

1XSTM-16 is the STM-16/OC-48 PIC not set to no-concatenate

4XSTM-16 is the STM-64/OC-192 PIC set to no-concatenate

1XSTM-64 is the STM-64/OC-192 PIC not set to no-concatenate

Table 20: SONET/SDH Framing Bytes for Specific Speeds

OH	STM-4	STM-16	STM-64	OC-12	OC-48	OC-192
A1	F6	F6	F6	F6	F6	F6
A2	28	28	28	28	28	28
J0/Z0	01/CC	01/CC	01/CC	—	—	—
C1	—	—	—	1..12	1..48	1..192
H1/H2	6A0A	6A0A	6A0A	620A	620A	620A
Concatenated mode	93FF	93FF	93FF	93FF	93FF	93FF

The use of J0 for framing has been deprecated. For DS-3 channels on a Channelized OC-12 interface, the bytes e1-quiet, bytes f1, bytes f2, bytes z3, and bytes z4 options have no effect. The bytes s1 option is supported only for channel 0; it is ignored if configured on channels 1 through 11. The bytes s1 value configured on channel 0 applies to all channels on the interface.

Configure SONET z0-increment Option

When configured in SDH framing mode, SONET/SDH interfaces on a Juniper Networks router might not interoperate with some older versions of ADMs or regenerators that require an incrementing STM ID. To resolve this incompatibility, you can explicitly configure an incrementing STM ID rather than a static one in the SDH overhead by including the z0-increment statement at the [edit interfaces *interface-name* sonet-options] hierarchy level. You should include this statement only for SDH mode; do not use it for SONET mode.

```
[edit interfaces so-fpc/pic/port sonet-options]
z0-increment;
```

To explicitly disable z0-incrementing, include the following statement:

```
[edit interfaces so-fpc/pic/port sonet-options]
no-z0-increment;
```

Current SDH standards specify a set of 3**n* overhead bytes in an STM-*n* that includes the J0 section trace byte. The rest are essentially unused (spare Z0) and contain hex values (0x01, 0xCC, 0xCC ... 0xCC). The older version of the standard specified that the same set of bytes should contain an incrementing sequence: 1, 2, 3, ..., 3**n*. Their use was still unspecified although they might have been used to assist in frame alignment.

The z0-increment option enables Juniper Networks routers to interoperate with older equipment that relies on those bytes for frame alignment.

The STM identifier has a precise definition in the SDH specifications. In ITU-T Recommendation G.707, *Network node interface for the synchronous digital hierarchy (SDH)* (03/96), section 9.2.2.2:

> NOTE: STM identifier: C1
>
> In earlier versions of the Recommendation, the content of bytes located at S (1, 7, 1) or [1, 6N + 1] to S (1,7, N) or [1, 7N] was defined as a unique identifier indicating the binary value of the multi-column, interleave depth coordinate, c. It may have been used to assist in frame alignment.

Configure the SONET Frame Checksum

By default, SONET interfaces use a 16-bit frame checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment may not support 32-bit checksums.

To configure a 32-bit checksum, include the fcs statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
fcs 32;
```

To return to the default 16-bit frame checksum, delete the fcs 32 statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options fcs 32
```

To explicitly configure a 16-bit checksum, include the fcs statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
fcs 16;
```

On a Channelized OC-12 interface, the SONET-options fcs statement is not supported. To configure FCS on each DS-3 channel, you must include the t3-options fcs statement in the configuration for each channel.

Configure SONET Loopback Capability

To configure loopback capability on a SONET interface, include the loopback statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
loopback (local | remote);
```

Packets can be looped on either the local or the remote router. To turn off loopback, remove the loopback statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options loopback
```

For DS-3 channels on a Channelized OC-12 interface, the SONET loopback statement is supported only for channel 0; it is ignored if included in the configuration for channels 1 through 11. The SONET loopback configured for channel 0 applies to all 12 channels equally. To configure loopbacks on the DS-3 channels, you must include the t3-options loopback statement in the configuration for each channel. Each DS-3 channel can be put in loopback mode independently.

Configure the SONET Path Trace Identifier

The SONET path trace identifier is a text string that identifies the circuit. If the string contains spaces, enclose it in quotation marks. The common convention is to use the circuit identifier as the path trace identifier. If you do not configure an identifier, the JUNOS software uses the system and interface names. The local system's path trace identifier is displayed when a show interfaces command is issued on the remote system.

For DS-3 channels on a Channelized OC-12 interface, you can configure a unique path trace for each of the 12 channels. Each path trace can be up to 16 bytes.

To configure a path trace identifier, include the path-trace statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
path-trace trace-string;
```

Configure SONET HDLC Payload Scrambling

SONET HDLC payload scrambling, which is enabled by default, provides better link stability. Both sides of a connection must either use or not use scrambling.

HDLC payload scrambling conflicts with traffic shaping configured using leaky bucket properties. If you configure leaky bucket properties, you must disable payload scrambling, because the software rejects configurations that have both features enabled. For more information, see "Configure Receive and Transmit Leaky Bucket Properties" on page 285.

On a Channelized OC-12 interface, the SONET payload-scrambler statement is ignored. To configure scrambling on the DS-3 channels on the interface, include the t3-options payload-scrambler statement in the configuration for each DS-3 channel.

To disable HDLC payload scrambling, include the no-payload-scrambler statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
no-payload-scrambler;
```

To return to the default, that is, to re-enable payload scrambling, delete the no-payload-scrambler statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options no-payload-scrambler
```

To explicitly enable payload scrambling, include the payload-scrambler statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
payload-scrambler;
```

Configure SONET RFC 2615 Support

RFC 2615 requires certain C2 header byte and Frame Checksum (FCS) settings that vary from the default values configured in accordance with RFC 1619. The newer values are optimized for stronger error detection, especially when combined with payload scrambling at higher bit rate links.

Table 21 shows the older (RFC 1619) and newer (RFC 2615) values, together with the Juniper Networks default values.

Table 21: SONET Default Settings

	RFC 1619	Default	RFC 2615
SONET C2 Header Byte	0XCF	0XCF	0X16
Frame Checksum (bit)	16	16	32
Payload Scrambling	n/a	Enabled	Enabled

To enable support for the RFC 2615 features, include the rfc-2615 statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
rfc-2615;
```

Configure APS

Automatic Protection Switching (APS) is used by SONET add/drop multiplexers (ADMs) to protect against circuit failures. The JUNOS implementation of APS allows you to protect against circuit failures between an ADM and one or more routers, and between multiple interfaces in the same router. When a circuit or router fails, a backup immediately takes over.

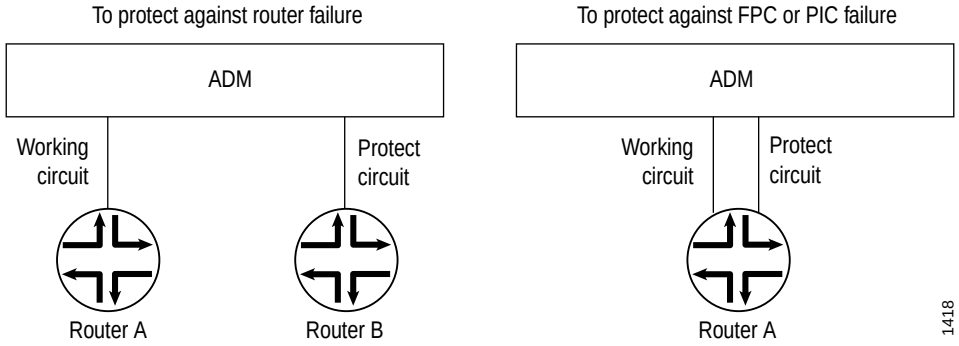
The JUNOS software supports APS 1 + 1 switching, bidirectional only, and either revertive or nonrevertive mode. The JUNOS software does not transmit identical data on the working and protect circuits, as the APS specification requires for 1 + 1 switching, but this causes no operational impact.

For DS-3 channels on a Channelized OC-12 interface, you can configure APS on channel 0 only. If you configure APS on channels 1 through 11, it is ignored.

With APS, you configure two circuits, a *working circuit* and a *protect circuit*. Normally, traffic is carried on the working circuit (that is, the working circuit is the active circuit), and the protect circuit is disabled. If the working circuit fails or degrades, or if the working router fails, the ADM and the protect router switch the traffic to the protect circuit, and the protect circuit becomes the active circuit.

To configure APS, you configure a *working* and a *protect* circuit, as shown in Figure 18. To protect against a router failure, you connect two routers to the ADM, configuring one of them as the working router and the second as the protect router. To protect against a PIC or an FPC failure, you connect one router to the ADM through both the working and protect circuits, configuring one of the PICs or FPCs as the working circuit and the second as the protect circuit.

Figure 18: APS Configuration Topologies



To configure APS, include the aps statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces interface-name sonet-options]
aps {
    advertise-interval milliseconds;
    authentication-key key;
    force;
    hold-time milliseconds;
    lockout;
    neighbor address;
    paired-group group-name;
    protect-circuit group-name;
    request;
    revert-time seconds;
    working-circuit group-name;
}
```

You can configure the following APS properties:

Configure Basic APS Support

To set up a basic APS configuration, configure one interface to be the working circuit and a second to be the protect circuit. If you are using APS to protect against router failure, configure one interface on each router. If you are using APS to protect against FPC failure, configure two interfaces on the router, one on each FPC.

For each working–protect circuit pair, configure the following:

Group name—Creates the association between the two circuits. Configure the same group name for both the working and protect routers.

Authentication key—You configure this on both interfaces. Configure the same key for both the working and protect routers.

Address of the other interface on the other router—If you are configuring one router to be the working router and a second to be the protect router, you must configure the address of the remote interface. You configure this on one or both of the interfaces.

The address you specify for the neighbor must never be routed through the interface on which APS is configured, or instability will result. APS neighbor only applies to inter-router configurations. We strongly recommend that you directly connect the working and protect routers and that you configure the interface address of this shared network as the neighbor address.

The working and protect configurations on the routers must match the circuit configurations on the ADM; that is, the working router must be connected to the ADM's working circuit and the protect router must be connected to the protect circuit.

To set up a basic APS configuration, include the following statements at the [edit interfaces *interface-name* sonet-options] hierarchy level:

On the working router/circuit:

```
[edit interfaces so-fpc/pic/port sonet-options]
aps {
    working-circuit group-name;
    authentication-key key;
    neighbor address; # Include only if protect circuit is on a different router
}
```

On the protect router/circuit:

```
aps {
    protect-circuit group-name;
    authentication-key key;
    neighbor address;# Include only if working circuit is on a different router
}
```

For example, configure Router A to be the working router and Router B to be the protect router.

On Router A (the working router):

```
[edit interfaces so-6/1/1 sonet-options]
aps {
    working-circuit San-Jose;
    authentication-key "$9$B2612345";
}
```

On Router B (the protect circuit):

```
[edit interfaces so-0/0/0 sonet-options]
aps {
    protect-circuit San-Jose;
    authentication-key "$9$B2612345";
    neighbor 192.168.1.2;# address of Router B's interface on the link between A and B
}
```

As a second example, configure one interface on a router to be the working circuit and another interface to be the protect circuit:

On Router A:

```
[edit interfaces so-2/1/1 sonet-options]
aps {
    working-circuit Hayward;
    authentication-key blarney;
}
[edit interfaces so-3/0/2 sonet-options]
aps {
    protect-circuit Hayward;
    authentication-key blarney;
}
```

Configure Switching between the Working and Protect Circuits

When there are multiple reasons to switch between the working and protect circuits, a priority scheme is used to decide which circuit to use. The routers and the ADM might automatically switch traffic between the working and protect circuits because of circuit and router failures. You can also choose to switch traffic manually between the working and protect circuits. There are three priority levels of manual configuration, listed here in order from lowest to highest priority:

Request (also known as manual switch)—Overridden by signal failures, signal degradations, or any higher-priority reasons.

Force (also known as forced switch)—Overrides manual switches, signal failures, and signal degradation.

Lockout (also known as lockout of protection)—Do not switch between the working and protect circuits.

A router failure is considered to be equivalent to a signal failure on a circuit.

To perform a manual switch, include the request statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level. This statement is honored only if there are no higher-priority reasons to switch.

```
[edit interfaces so-fpc/pic/port sonet-options aps]
request (protect | working);
```

When the working circuit is operating in nonrevertive mode, use the request working statement to switch the circuit manually to being the working circuit or to override the revert timer.

To perform a forced switch, include the force statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level. This statement is honored only if there are no higher-priority reasons to switch. This configuration can be overridden by a signal failure on the protect circuit, thus causing a switch to the working circuit.

```
[edit interfaces so-fpc/pic/port sonet-options aps]
force (protect | working);
```

To configure a lockout of protection, forcing the use of the working circuit and locking out the protect circuit regardless of anything else, include the lockout statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
lockout;
```

Configure Revertive Mode

By default, APS is nonrevertive, which means that if the protect circuit becomes active, traffic is not switched back to the working circuit unless the protect circuit fails or you manually configure a switch to the working circuit. In revertive mode, traffic is automatically switched back to the working circuit.

You should configure the ADM and routers consistently with regard to revertive or nonrevertive mode.

To configure revertive mode, include the revert-time statement, specifying the amount of time to wait after the working circuit has again become functional before making the working circuit active again:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
revert-time seconds;
```

If you are using nonrevertive APS, you can use the request working statement to switch the circuit manually to being the working circuit or to override the revert timer (configured with the revert-time statement).

Configure APS Timers

The protect and working routers periodically send packets to their neighbors to advertise that they are operational. By default, these advertisement packets are sent every 1000 milliseconds. A router considers its neighbor to be operational for a period, called the hold time, that is, by default, three times the advertisement interval. If the protect router does not receive an advertisement packet from the working router within the hold time configured on the protect router, the protect router assumes that the working router has failed and becomes active.

APS is symmetric; either side of a circuit can time out the other side (for example, when detecting a crash of the other). Under normal circumstances, the failure of the protect router does not cause any changes because the traffic is already moving on the working router. However, if you had configured request protect and the protect router failed, the working router would enable its interface.

To modify the advertisement interval, include the advertise-interval at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
advertise-interval milliseconds;
```

To modify the hold time, include the hold-time at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
hold-time milliseconds;
```

The advertisement intervals and hold times on the protect and working routers can be different.

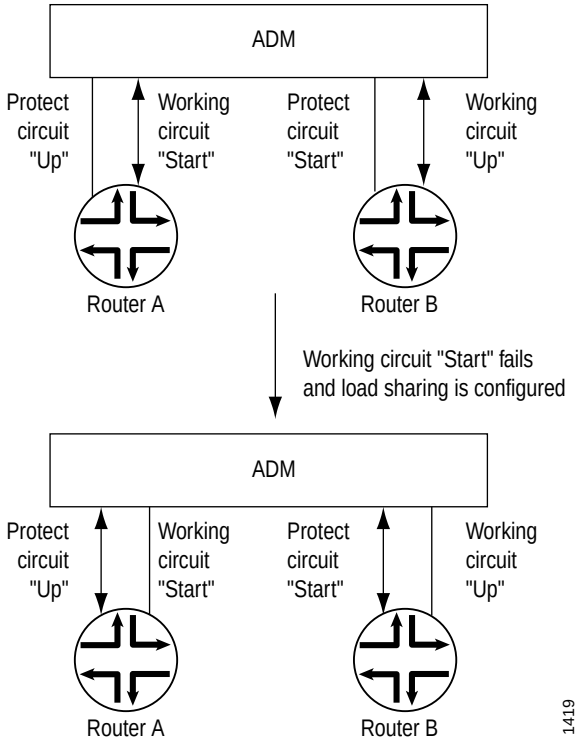
Configure APS Load Sharing between Circuit Pairs

When two routers are connected to a single ADM, you can have them back up each other on two different pairs of circuits. This arrangement provides load balancing between the routers if one of the working circuits fails.

Figure 19 illustrates load sharing between circuits on two routers. Router A has a working circuit “Start” and a protect circuit “Up,” and Router B has a working circuit “Up” and a protect circuit “Start.” Under normal circumstances, Router A carries the “Start” circuit traffic and Router B carries the “Up” circuit traffic. If the working circuit “Start” were to fail, Router B would end up carrying all the traffic for both the “Start” and “Up” circuits.

To balance the load between the circuits, you pair the two circuits. In this case, you pair the "Start" and "Up" circuits. Then, if the working circuit "Start" fails, the two routers automatically switch the "Up" traffic from the working to the protect circuit so that each router is still carrying only one circuit's worth of traffic. That is, the working circuit on Router A would be "Up" and the working circuit on Router B would be "Start."

Figure 19: APS Load Sharing between Circuit Pairs



To configure load sharing between two working–protect circuit pairs, include the paired-group statement when configuring one of the circuits on one of the routers. In this statement, the *group-name* is the name of the group you assigned to one of the circuits with the working-circuit and protect-circuit statements. The software automatically configures the remainder of the load-sharing setup based on the group name.

```
[edit interfaces so-fpc/pic/port sonet-options aps]
paired-group group-name;
```

Example: Configure APS Load Sharing between Circuit Pairs

Configure APS load sharing to match the configuration shown in Figure 19:

On Router A:

```
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set working-circuit start
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set authentication-key linsey
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
...
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set protect-circuit up
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set authentication-key woolsey
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
```

On Router B:

```
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set working-circuit up
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set authentication-key woolsey
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
...
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set protect-circuit start
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set authentication-key linsey
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
```

Configure the Media MTU

The default media MTU size used on a physical interface depends on the encapsulation being used on that interface. Table 3 on page 41 and Table 6 on page 43 list MTU sizes for each encapsulation type. For information about configuring the encapsulation on an interface, see “Configure Interface Encapsulation” on page 287.

To modify the default media MTU size for a physical interface, include the mtu statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
mtu bytes;
```

If you change the size of the media MTU, you must ensure that the size is equal to or greater than the sum of the protocol MTU and the encapsulation overhead. You configure the protocol MTU by including the mtu statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level, as discussed in “Set the Protocol MTU” on page 131.

Configure the Clock Source

For interfaces such as SONET that can use different clock sources, you can configure the source of the transmit clock on each interface. The source can be internal (also called line timing or normal) or external (also called loop timing). The default source is internal, which means that each interface uses the router's internal stratum 3 clock.

For DS-3 channels on a Channelized OC-12 interface, the clocking statement is supported only for channel 0; it is ignored if included in the configuration of channels 1 through 11. The clock source configured for channel 0 applies to all channels on the Channelized OC-12 interface. The individual DS-3 channels use a gapped 45-MHz clock as the transmit clock.

On Channelized STM-1 interfaces, you should configure the clock source at one side of the connection to be internal (the default JUNOS configuration) and configure the other side of the connection to be external.

To configure loop timing on an interface, include the clocking external statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
clocking external;
```

To explicitly configure line timing on an interface, include the clocking internal statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
clocking internal;
```

Configure Receive and Transmit Leaky Bucket Properties

Congestion control is particularly difficult in high-speed networks with high volumes of traffic. When congestion occurs in such a network, it is usually too late to react. You can avoid congestion by regulating the flow of packets into your network. Smoother flows prevent bursts of packets from arriving at (or being transmitted from) the same interface and causing congestion.

For all interface types except ATM, Fast Ethernet, and Gigabit Ethernet, you can configure leaky bucket properties, which allow you to limit the amount of traffic received on and transmitted by a particular interface. You effectively specify what percentage of the interface's total capacity can be used to receive or transmit packets. You might want to set leaky bucket properties to limit the traffic flow from a link that is known to transmit high volumes of traffic.

Instead of configuring leaky bucket properties, you can limit traffic flow by configuring policers. Policers work on all interfaces. For more information, see the *JUNOS Internet Softw are Configur ation Guide: P olicy Frame work*.

The leaky bucket is used at the host-network interface to allow packets into the network at a constant rate. Packets might be generated in a bursty manner, but after they pass through the leaky bucket, they enter the network evenly spaced. In some cases, you might want to allow short bursts of packets to enter the network without smoothing them out. By controlling the number of packets that can accumulate in the bucket, the threshold property controls burstiness. The maximum number of packets entering the network in t time units is threshold + rate * t.

By default, leaky buckets are disabled and the interface can receive and transmit packets at the maximum line rate.

HDLC payload scrambling conflicts with traffic shaping configured using leaky bucket properties. If you configure leaky bucket properties, you must disable payload scrambling, because the software rejects configurations that have both features enabled. For more information, see "Configure SONET HDLC Payload Scrambling" on page 276.

For each DS-3 channel on a Channelized OC-12 interface, you can configure unique receive and transmit buckets.

To configure leaky bucket properties, include one or both of the receive-bucket and transmit-bucket statements at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
receive-bucket {
    overflow (tag | discard);
    rate percentage;
    threshold number;
}
transmit-buckets {
    overflow (discard);
    rate percentage;
    threshold number;
}
```

In the rate option, specify the percentage of the interface line rate that is available to receive or transmit packets. The percentage can be a value from 0 (none of the interface line rate is available) to 100 (the maximum interface line rate is available). For example, when you set the line rate to 33, the interface receives or transmits at one third of the maximum line rate.

In the threshold option, specify the bucket threshold, which controls the burstiness of the leaky bucket mechanism. The larger the value, the more bursty the traffic, which means that over a very short amount of time the interface can receive or transmit close to line rate, but the average over a longer time is at the configured bucket rate. The threshold can be a value from 0 through 16777215 bytes. For ease of entry, you can enter *number* either as a complete decimal number or as a decimal number followed by the abbreviation k (1,000) or m (1,000,000). For example, the entry threshold 2m corresponds to a threshold of 2,000,000 bytes.

In the overflow option, specify how to handle packets that exceed the threshold:

tag—(receive-bucket only) Tag, count, and process received packets that exceed the threshold.

discard—Discard received packets that exceed the threshold. No counting is done.

Damp Interface Transitions

By default, when an interface changes from being up to being down, or from down to up, this transition is advertised immediately to the router software and hardware. In some situations, for example, when an interface is connected to an ADM or WDM, or to protect against SONET framer holes, you might want to damp interface transitions, thereby not advertising the interface's transition until a certain period of time has transpired. When you have damped interface transitions and the interface goes from up to down, the interface is not advertised to the rest of the system as being down until it has remained down for the hold-time period. Similarly when an interface goes from down to up, it is not advertised as being up until it has remained up for the hold-time period.

To damp interface transitions, include the hold-time statement at the [edit interfaces *interface-name*] hierarchy level:

```
hold-time up milliseconds down milliseconds;
```

The time can be a value from 0 through 65,534 milliseconds. The time value that you specify is rounded up to the nearest whole second. The default value is 0, which means that interface transitions are not damped.

Configure Interface Encapsulation

Point-to-Point Protocol (PPP) encapsulation is the default encapsulation type for physical interfaces. You need not configure encapsulation for any physical interfaces that support PPP encapsulation. If you do not configure encapsulation, PPP is used by default. For physical interfaces that do not support PPP encapsulation, you must configure an encapsulation to use for packets transmitted on the interface. You can optionally configure an encapsulation on a logical interface, which is the encapsulation used within certain packet types.

Configure the Encapsulation on a Physical Interface

For SONET/SDH interfaces, the physical interface encapsulation can be one of the following:

Point-to-Point Protocol (PPP)—PPP encapsulation is defined in RFC 1331, *The Point-to-Point Protocol (PPP) for the Transmission of Multiprotocol Datagrams over Point-to-Point Links*. PPP is the default encapsulation type for physical interfaces. Two related versions are supported:

Circuit cross-connect (CCC) version (ppp-ccc)—The logical interfaces do not require an encapsulation statement. When you use this encapsulation, you can configure the family ccc only.

Translational cross-connect (TCC) version (ppp-tcc)—Similar to CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

Cisco HDLC—E1, E3, SONET, T1, and T3 interfaces can use Cisco HDLC encapsulation. Two related versions are supported:

CCC version (cisco-hdlc-ccc)—The logical interfaces do not require an encapsulation statement. When you use this encapsulation, you can configure the family ccc only.

TCC version (cisco-hdlc-tcc)—Similar to CCC and has the same configuration restrictions, but is used for circuits with different media on either side of the connection.

Frame Relay—Defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. E1, E3, SONET, T1, and T3 interfaces can use Frame Relay encapsulation. Two related versions are supported:

CCC version (frame-relay-ccc)—The same as standard Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to CCC, and the logical interface must also have frame-relay-ccc encapsulation.

TCC version (frame-relay-tcc)—Similar to Frame Relay CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

To configure the encapsulation on a physical interface, include the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
encapsulation (cisco-hdlc | cisco-hdlc-ccc | cisco-hdlc-tcc | frame-relay | frame-relay-ccc |
frame-relay-tcc | frame-relay-tcc | ppp | ppp-ccc | ppp-tcc);
```

When you configure a point-to-point encapsulation (such as PPP or Cisco HDLC) on a physical interface, the physical interface can have only one logical interface (that is, only one unit statement) associated with it. When you configure a multipoint encapsulation (such as Frame Relay), the physical interface can have multiple logical units, and the units can be either point to point or multipoint. Use PPP if you are running Cisco IOS Release 12.0 or later. If you need to run Cisco HDLC, the JUNOS software automatically configures an ISO family MTU of 4469 in the router. This is due to an extra byte of padding used by Cisco.

For more information about physical interface encapsulation, see “Configure the Encapsulation on a Physical Interface” on page 44.

Example: Configure the Encapsulation on a Physical Interface

Configure PPP encapsulation on a SONET interface. The second two family statements allow IS-IS and MPLS to run on the interface.

```
[edit interfaces]
so-7/0/0 {
    encapsulation ppp;
    unit 0 {
        point-to-point;
        family inet {
            address 192.168.1.113/32 {
                destination 192.168.1.114;
            }
        }
        family iso;
        family mpls;
    }
}
```

Configure the Encapsulation on a Logical Interface

Generally, you configure an interface's encapsulation at the [edit interfaces *interface-name*] hierarchy level. However, for Frame Relay encapsulation, you also can configure the encapsulation type that is used inside the Frame Relay packet itself. To do this, include the encapsulation statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level, specifying the frame-relay-ccc or frame-relay-tcc option:

```
[edit interfaces interface-name unit logical-unit-number]
encapsulation (frame-relay-ccc | frame-relay-tcc);
```

The ATM encapsulations are defined in RFC 1483, *Multiprotocol Encapsulation o ver ATM Adaptation Layer 5*.

With the atm-nlpid, atm-cisco-nlpid, and atm-vc-mux encapsulations, you can configure the family inet only. With the circuit cross-connect (CCC) encapsulations, you cannot configure a family on the logical interface. A logical interface cannot have frame-relay-ccc encapsulation unless the physical device also has frame-relay-ccc encapsulation. A logical interface cannot have frame-relay-tcc encapsulation unless the physical device also has frame-relay-tcc encapsulation. In addition, you must assign this logical interface a DLCI in the range 512 through 1022 and configure it as point-to-point.

For more information about logical interface encapsulation, see "Configure the Encapsulation on a Logical Interface" on page 106.

Example: Configure SONET Interfaces

SONET interfaces can use either PPP or Cisco HDLC encapsulation. Use PPP if you are running Cisco IOS Release 12.0 or later. If you need to run Cisco HDLC, the JUNOS software automatically configures an ISO family MTU of 4469 in the router. This is due to an extra byte of padding used by Cisco. The following configuration, which uses PPP encapsulation, is sufficient to get a SONET OC-3 or OC-12 interface up and running:

```
[edit interfaces]
   so-fpc/pic/port {
      encapsulation ppp;
      unit 0 {
         family inet {
            address local-address {
               destination remote-address;
            }
         }
      }
   }
```

Configure Aggregated SONET/SDH Interfaces

The JUNOS software enables link aggregation of SONET/SDH interfaces; this is similar to Ethernet link aggregation, but is not defined in a public standard. You configure an aggregated SONET/SDH virtual link by specifying the link number as a physical device and then associating a set of physical interfaces that have the same speed.

The JUNOS software does not provide load balancing for multicast traffic on aggregated interfaces. If a link carrying multicast data goes down, another link carries the traffic. This provides redundancy, not more bandwidth.

By default, no aggregated SONET/SDH interfaces are created. You must define the number of aggregated SONET/SDH interfaces by including the device-count statement at the [edit chassis aggregated-devices sonet] hierarchy level:

```
[edit chassis aggregated-devices sonet]
   device-count number;
```

The maximum number of aggregated interfaces is 16, and the assigned number can range from 0 through 15. For more information, see the *JUNOS Internet Softw are Guide: Getting Started*.

SONET aggregation is proprietary to the JUNOS software and might not work with other software.

To configure aggregated SONET/SDH interfaces, assign a number for the aggregated SONET/SDH interface as*x* at the [edit interfaces] hierarchy level:

```
[edit interfaces]
asx {
...
}
```

The following example shows an aggregated SONET/SDH configuration:

```
[edit interfaces]
as0 {
    aggregated-sonet-options {
        minimum-links 1;
        link-speed oc3;
    }
    unit 0 {
        family inet {
            address 10.2.11.1/32 {
                destination 10.2.11.3;
            }
        }
    }
}
```

You also need to specify the constituent physical interfaces by including the aggregate statement at the [edit interfaces *interface-name* sonet-options] hierarchy level; for more information, see “Configure SONET Link Aggregation” on page 292. You can optionally specify other physical properties that apply specifically to the aggregated SONET interfaces; for details, see “Configure SONET/SDH Physical Interface Properties” on page 272. For a sample configuration, see “Example: Configure Aggregated SONET/SDH Interfaces” on page 294.

To remove the configuration statements related to as*x* and set the aggregated SONET/SDH interface to down state, delete the interface from the configuration:

```
[edit]
user@host# delete interfaces asx
```

However, the aggregated SONET/SDH interface is not deleted until you delete the chassis aggregated-devices sonet device-count configuration statement.

You can configure the following aggregated SONET/SDH properties:

Configure SONET Link Aggregation on page 292

Configure Aggregated SONET Link Speed on page 292

Configure Aggregated SONET Minimum Links on page 292

Configure Filters or Sampling on Aggregated SONET Links on page 293

Configure SONET Link Aggregation

On SONET/SDH interfaces, you can associate a physical interface with an aggregated SONET interface. To associate the interface with an aggregated SONET link, include the aggregate statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces interface-name sonet-options]
aggregate asx;
```

x is the interface instance number and can range from 0 through 15, for a total of 16 aggregated interfaces. You should not mix SONET and SDH mode on the same aggregated interface. You must also include a statement configuring as*x* at the [edit interfaces] hierarchy level. For a sample configuration, see "Example: Configure Aggregated SONET/SDH Interfaces" on page 294.

You can combine like interfaces only, so each physical interface in the aggregate has to be the same speed.

Configure Aggregated SONET Link Speed

On aggregated SONET interfaces, you can set the required link speed for all interfaces included in the bundle. All interfaces that make up a bundle must be the same speed. If you include in the aggregated SONET interface an individual link that has a speed different from the speed you specify in the link-speed parameter, an error message will be logged. To set the required link speed, include the link-speed statement at the [edit interfaces *interface-name* aggregated-sonet-options] hierarchy level:

```
[edit interfaces interface-name aggregated-sonet-options]
link-speed speed;
```

speed can be one of the following values:

oc3—Links are OC-3c or STM-1c.

oc12—Links are OC-12c or STM-4c.

oc48—Links are OC-48c or STM-16c.

oc192—Links are OC-192c or STM-64c.

Configure Aggregated SONET Minimum Links

On aggregated SONET interfaces, you can set the minimum number of links that must be up for the bundle as a whole to be labeled up. To set the minimum number, include the minimum-links statement at the [edit interfaces *interface-name* aggregated-sonet-options] hierarchy level:

```
[edit interfaces interface-name aggregated-sonet-options]
minimum-links number;
```

By default, minimum-link has a value of 1. *number* can be a value from 1 through 8.

Configure Filters or Sampling on Aggregated SONET Links

To set up firewall filters or sampling on aggregated SONET interfaces, you must configure the as*x* interface with these properties. The filters function in the same manner as on other interfaces.

To configure a filter, include the filter statement at the [edit interfaces as*x*] hierarchy level:

```
[edit interfaces]
asx {
    unit 0 {
        family inet {
            address 10.2.11.1/32 {
                destination 10.2.11.3;
            }
            filter {
                input input-filter-name;
                output output-filter-name;
            }
        }
    }
}
```

You must also configure separate statements that define the properties of the filter. The following is a generic example:

```
[edit firewall]
filter input-filter-name {
    term match-any-input {
        then {
            accept;
        }
    }
}
filter output-filter-name {
    term match-any-output {
        then {
            accept;
        }
    }
}
```

For more information, see the *JUNOS Internet Softw are Configur ation Guide: P olicy Frame work*.

You configure sampling on aggregated SONET/SDH interfaces in a similar way, as shown in this example:

```
[edit interfaces]
asx {
    unit 0 {
        family inet {
            address 10.2.11.1/32 {
                destination 10.2.11.3;
            }
            filter {
                input input-sampler-name;
            }
        }
    }
}
```

You define the sampling filter and the forwarding action as in the following examples:

```
[edit firewall]
filter input-sampler-name {
    term match-any-input {
        then {
            sample;
            accept;
        }
    }
}

[edit forwarding-options]
sampling {
    input {
        family inet {
            rate 10000;
            run-length 1;
        }
    }
}
```

For more information, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

Example: Configure Aggregated SONET/SDH Interfaces

The following configuration is sufficient to get an aggregated SONET/SDH interface up and running:

```
[edit interfaces]
as0 {
    aggregated-sonet-options {
        minimum-links 1;
        link-speed oc3;
    }
    unit 0 {
        family inet {
            address 10.2.11.1/32 {
                destination 10.2.11.3;
            }
        }
    }
}

[edit chassis]
aggregated-devices {
    sonet {
        device-count 15;
    }
}

[edit interfaces]
so-1/3/0 {
    sonet-options {
        aggregate as0;
    }
}
```